Blockchain for documents- the future of security

By Ross Frank

What is Blockchain

Traditional database

- Stored in a structured way
  - Tables (relational database)
    - Data tables

ID	First Name	Last Name	Email	Year of Birth
1	Peter	Lee	plee@university.edu	1992
2	Jonathan	Edwards	jedwards@university.edu	1994
3	Marilyn	Johnson	mjohnson@university.edu	1993
6	Joe	Kim	jkim@university.edu	1992
12	Haley	Martinez	hmartinez@university.edu	1993
14	John	Mfume	jmfume@university.edu	1991
15	David	Letty	dletty@university.edu	1995

Table: Students

Tutor	Student	Course
14	1	Algebra
1	12	Algebra
12	2	Algebra
2	15	Algebra
14	3	Statistics
3	15	Statistics

Table: tutorship

    - Eliminate redundancy by linking data through ID = relations

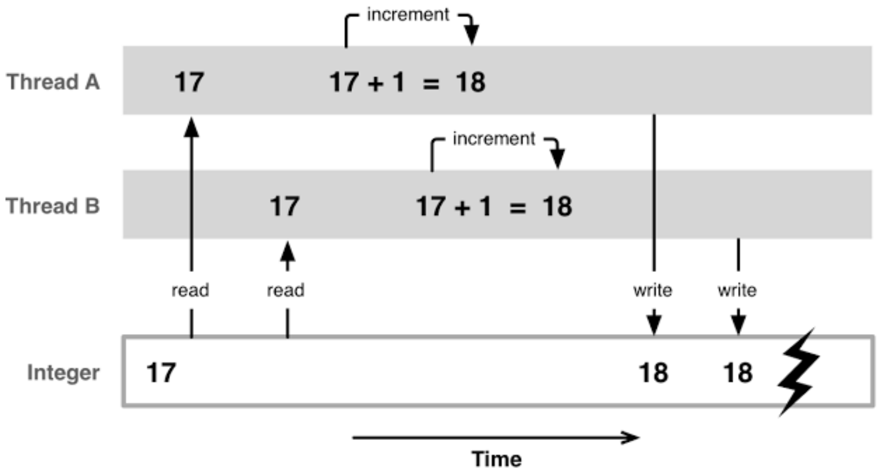
Problems

- Stored in one place
  - Fail-over mechanisms are copies of the database
  - Any copies must sync back to a master copy
  - Copies cannot easily accept new entries
    - Must sync updates back to master ASAP

- Data integrity
  - Hard to keep track of all changes
  - Entries can change without warning
    - unless specifically programmed to retain history
    - but that history can then again be edited etc

Problems

- Concurrent modifications
  - Race condition
    - Multiple users changing the same value
  - Edit wars



Definition

- Wikipedia !
  - A **blockchain** is a distributed database that maintains a continuously growing list of records, called blocks, secured from tampering and revision.
  - A block is a collection of transactions that are added to the chain

- Data security
  - Users can have copy of the database
    - For integrity checks & fail-over security
    - Majority of user decides which data is authoritative

Definition

- Implementation details
  - Each block contains a **timestamp** and a **link** to a previous block.

  - Guarantee of authorship & integrity
    - Use of cryptographic concepts
      - Hashing & digital signatures

  - By design, blockchains are inherently resistant to modification of the data
    - The main intention is to always store any modification in a new record
    - no overwriting or erasing
    - once recorded, the data in a block cannot be altered retroactively

Why?

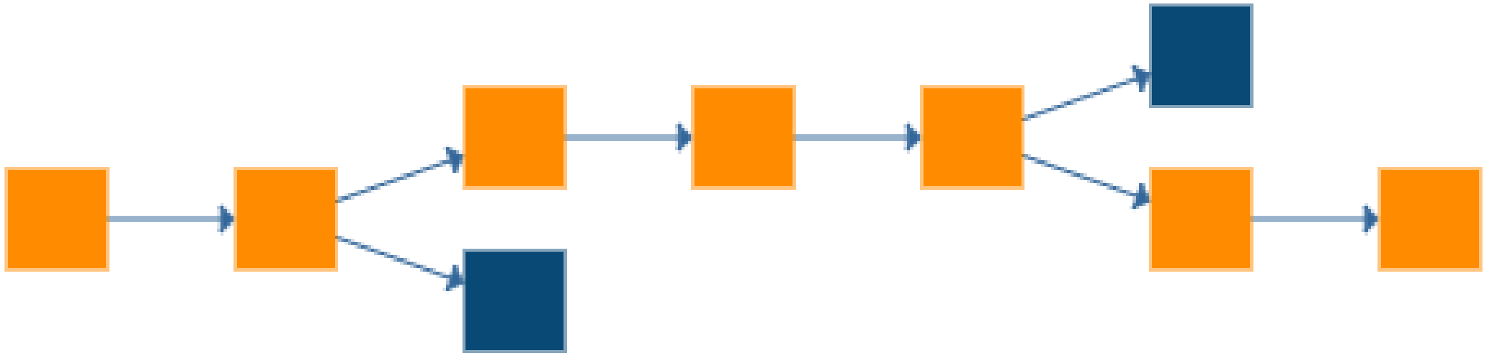
- Integrity
  - “The document has this exact content.”
- Authentication
  - *“I created this document. And I can prove it.”*
- Non-repudiation
  - *“He created this document. And I can prove it.”*

Why?

- *"Hey, I’ve created this hash on 10 Oct 2016: here is the transaction in the blockchain which contains the hash. I’ve created it according to this formula from this file.”*

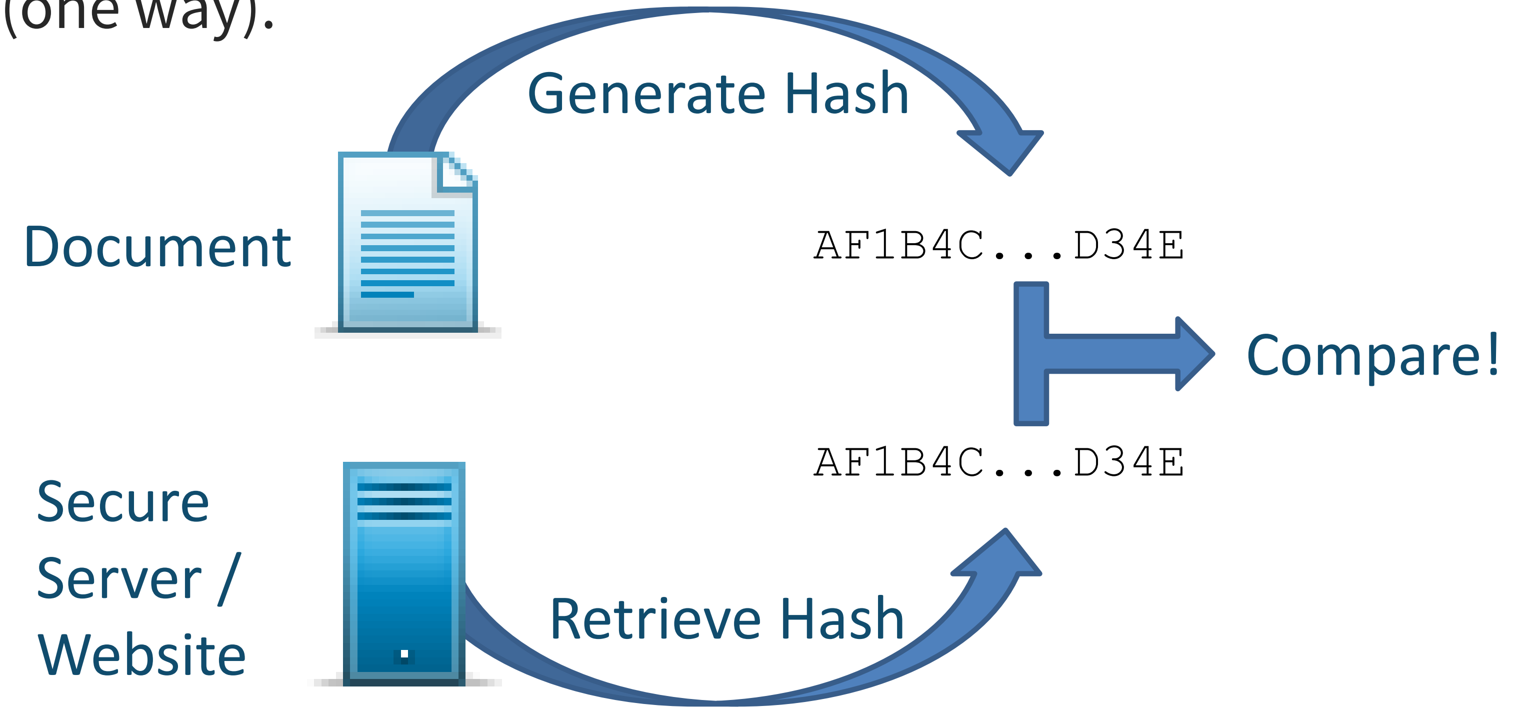
  - Integrity
  - Authentication
  - Non-repudiation
  - Timestamp

Basic concept



Hashing

- Turns an arbitrary block of data into a fixed-size bit string.
- Used for verification of data integrity.
  - Any small change to input has huge effect on hash value.
- Non-reversible (one way).



Encryption

- Using two separate but compatible keys to encrypt information.
  - Encrypt data

  - Sign data

- Can be decrypted => two-way.

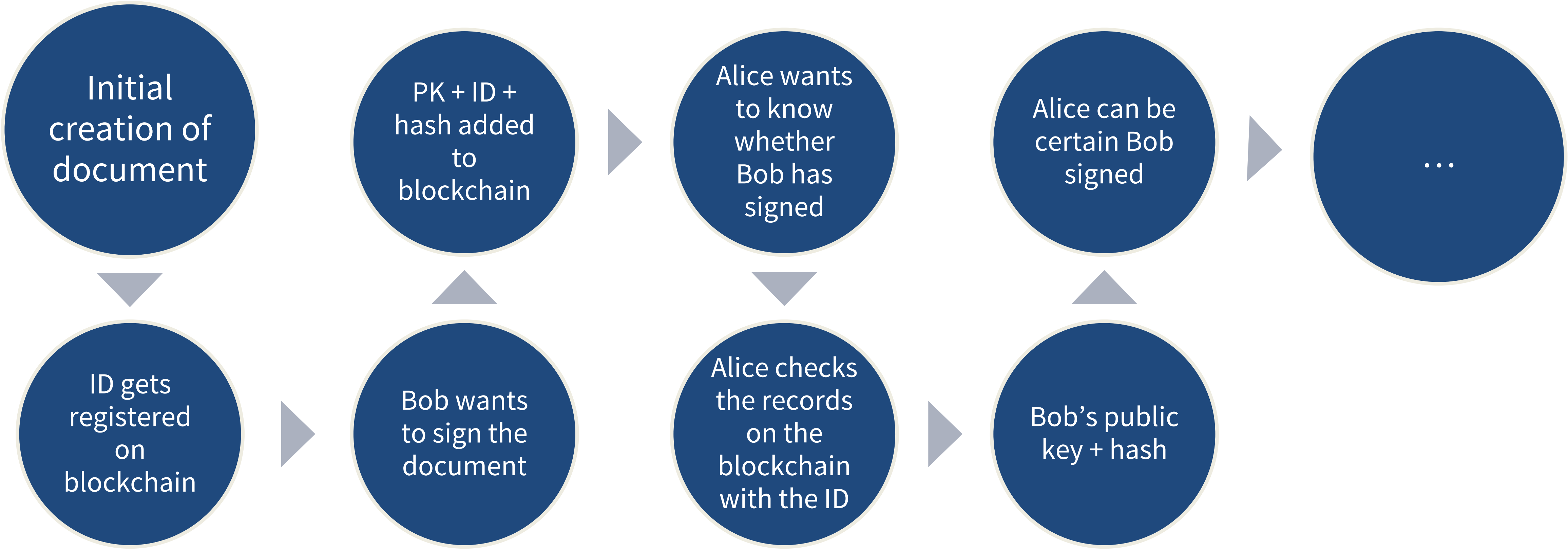
Relation to pdf

- Pdf documents can be digitally signed.
  - Requires Certificate Authority (centralized).
  - Requires timeserver (centralized).
  - Can not be signed in parallel.
  - Signatures live in the document.

Opportunities

- Data in a blockchain
  - Can be signed using known infrastructure.
  - Is automatically validated and timestamped.
  - Can be viewed by everyone.
  - Can live separately from the physical (real world) data it references.

Our idea - high level



Our idea - detail level

- Store meta-information of the pdf document on a blockchain:
  - ID,
  - hash (+ algorithm),
  - signature (+ algorithm),
  - fields that can be chosen by the end-user.
    - E.g. “currently awaiting feedback”, “asset has been checked by customs USA”, etc.

Web of trust

Web of trust

- In cryptography, a web of trust is a concept to establish the authenticity of the binding between a public key and its owner.

  Source: Wikipedia

Web of trust

- Bob can look up the public key of Alice
  - assuming public keys are truly publicly available,
  - or Alice can simply give Bob her public key.

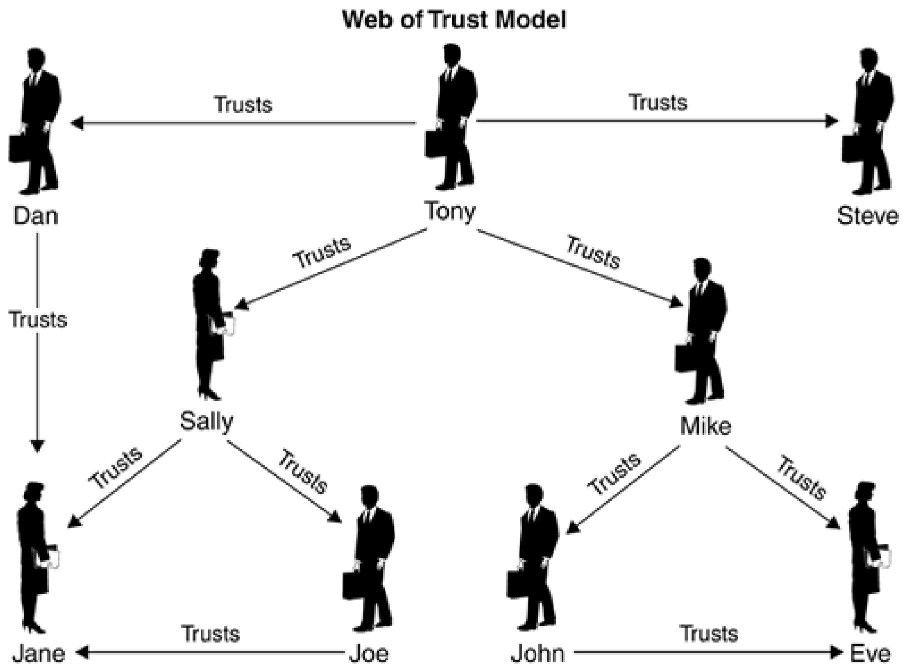
- Bob signs the public key of Alice with his private key.

- Other users can see all these records.
  - They can verify (using Bob's public key) that Bob has signed Alice's key.
  - This is considered as "Bob trusts Alice".

ITEXT

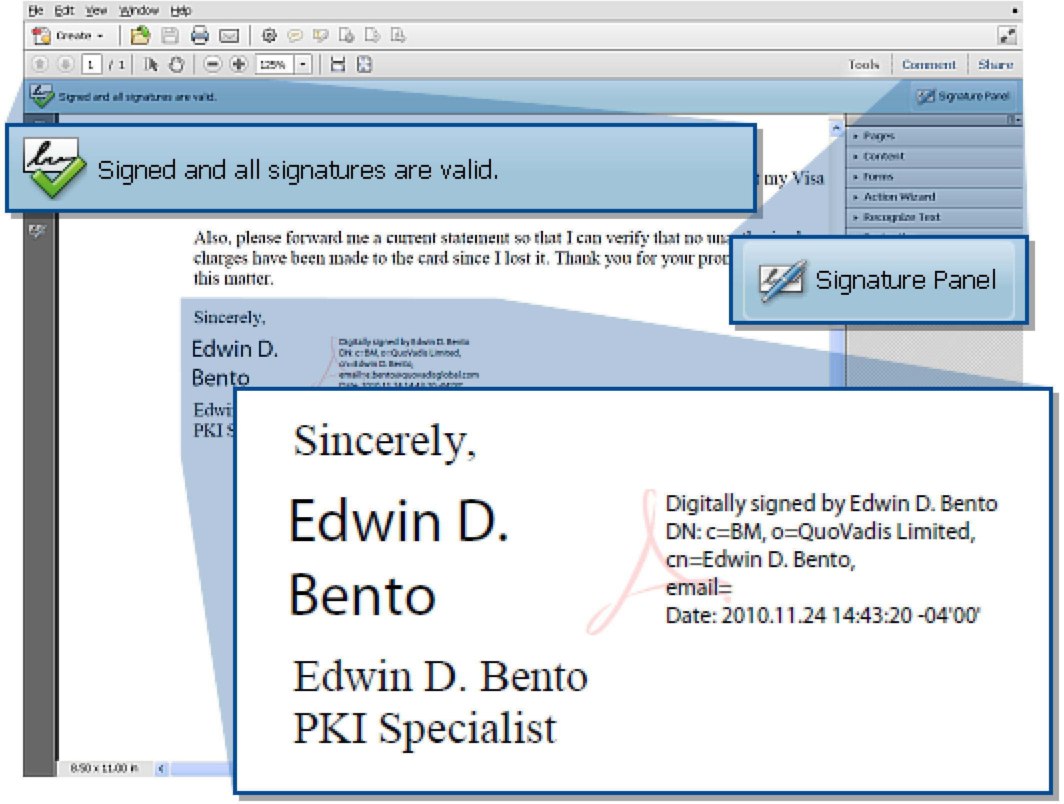
Web of trust

- This builds a graph where some nodes are connected by a “X trusts Y” relationship.
- The application built on top of this framework can then decide how to handle trusted vs. untrusted users.



Web of trust

- Example

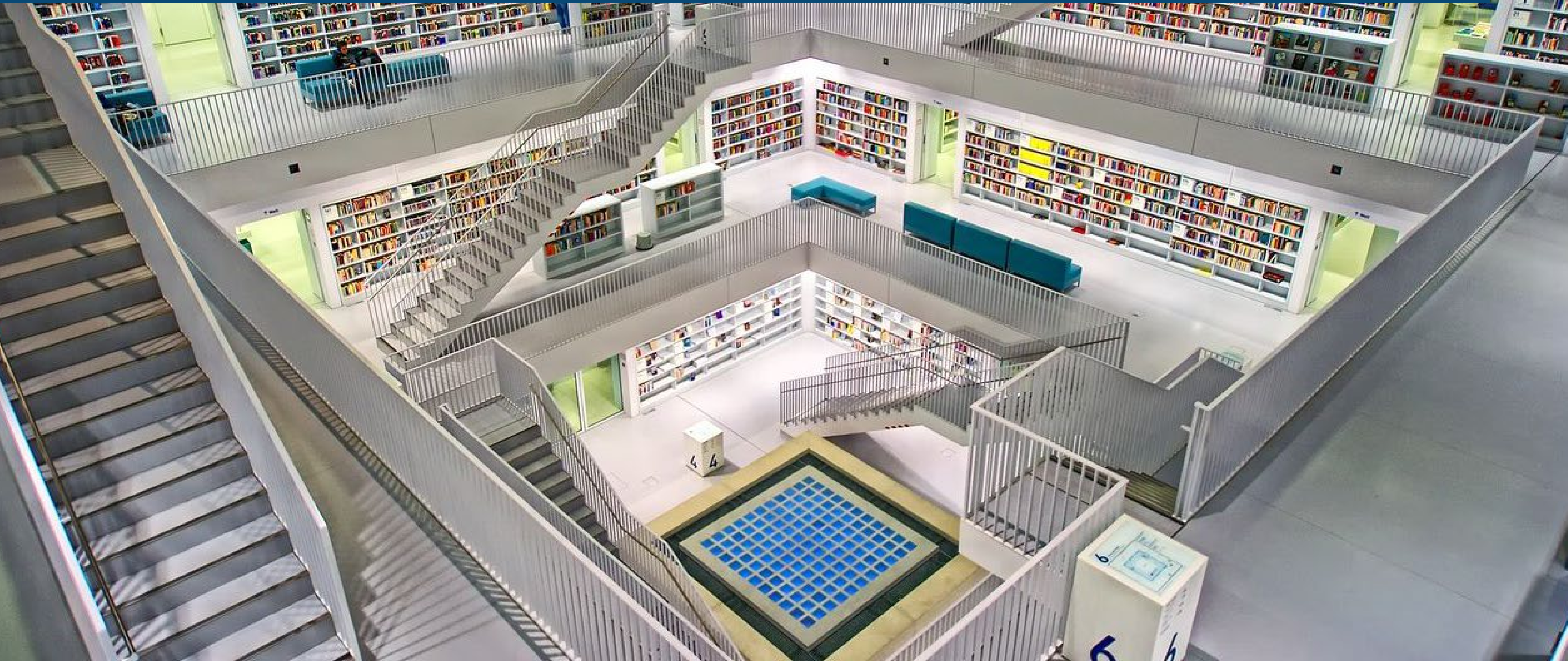


Web of trust

- Extensions are possible.
- Add a status field “ACK” or “NACK”.
  - Now Bob can revoke his trust in Alice.
  - Only the most recent record is taken into account.
- Allows temporary trust (interim workers).

Use case(s)

Blockchain for documents



Document record

Document ID: [<ABCDEF>, <ABCDEF>]

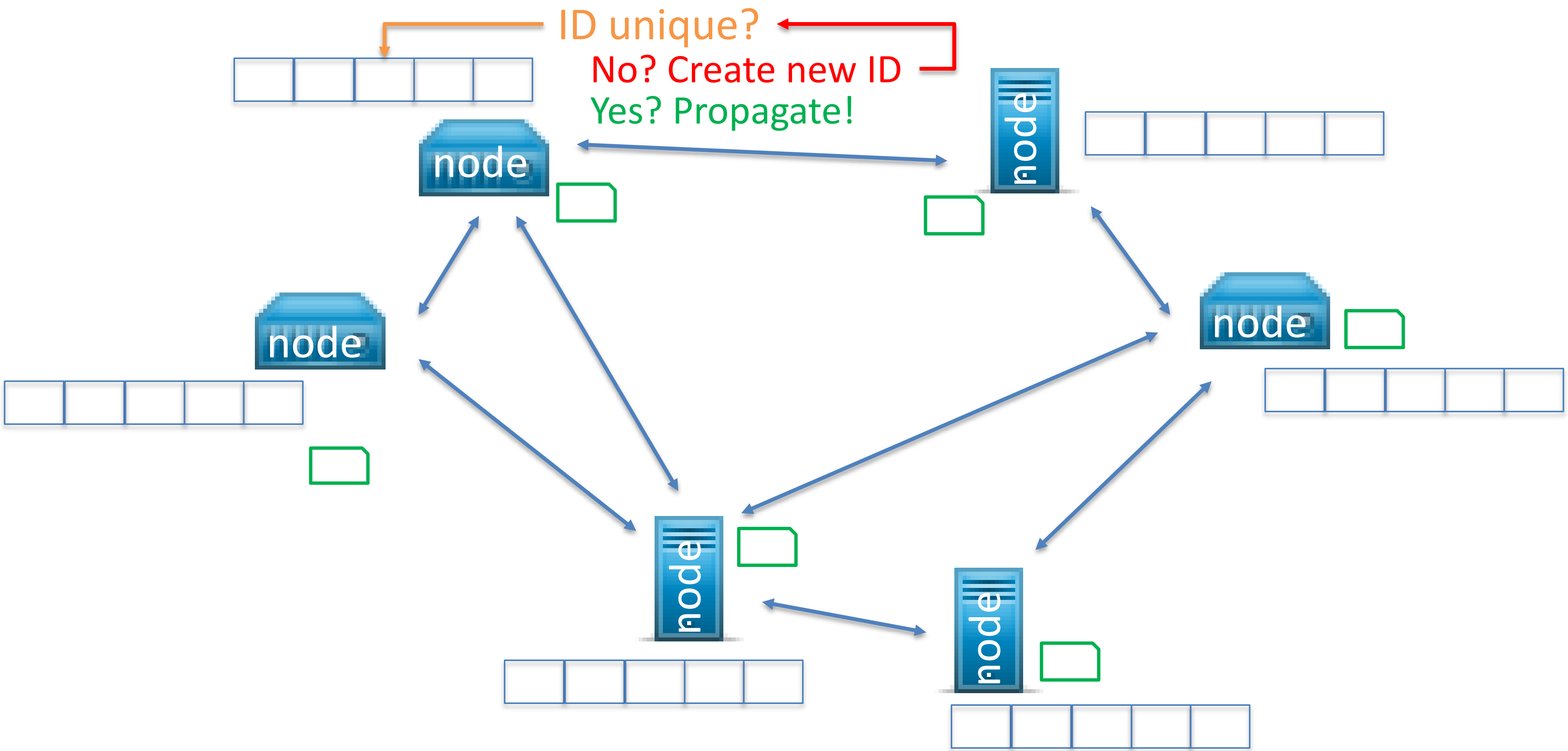
Timestamp

Signed — Document hash

Certificate of signer
- Identity
- Public key

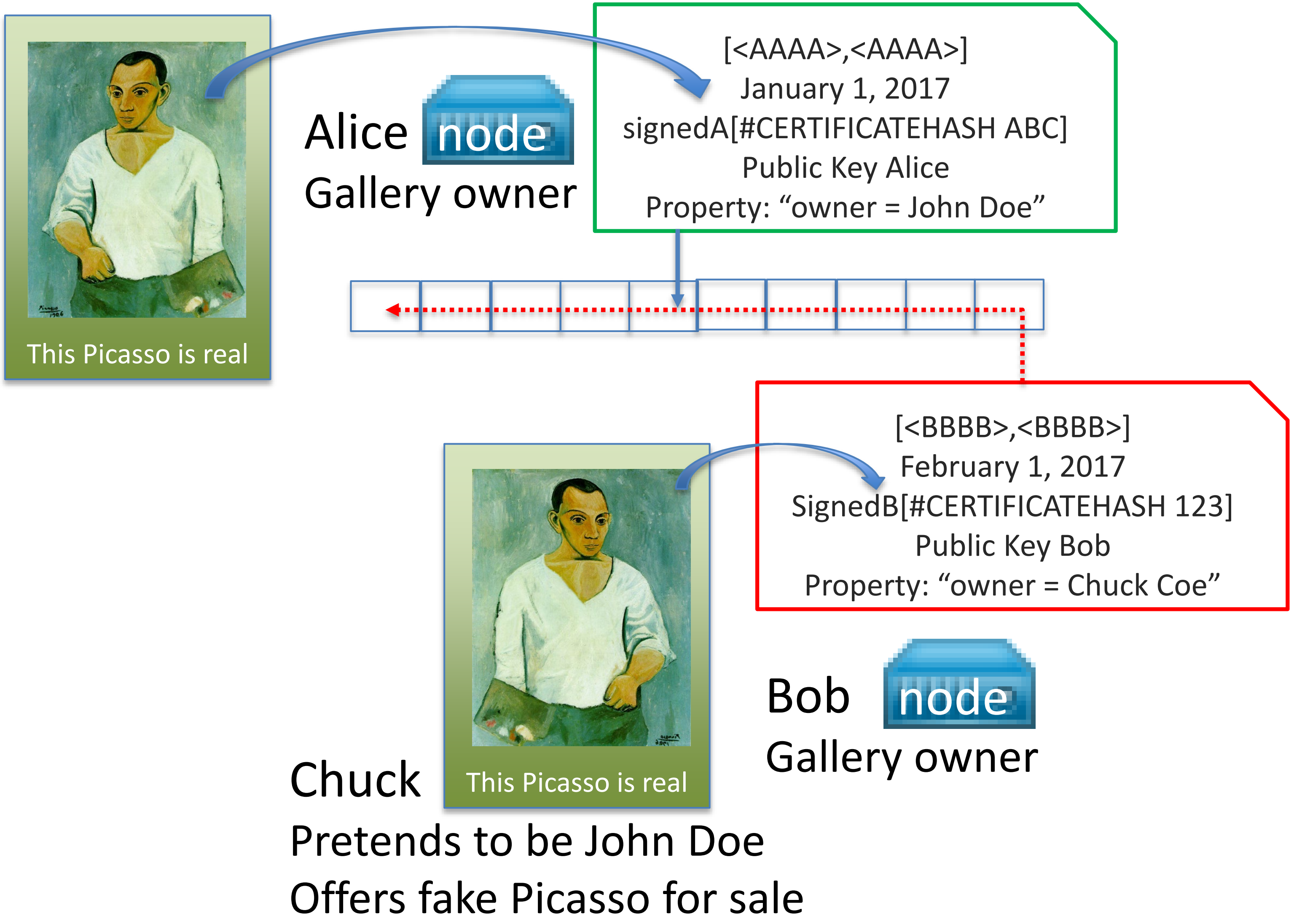
Compressed property list

Records are distributed



Use case: certificate of authenticity

Use case: certificate of authenticity



1st attempt to offer a forged painting with a fake certificate fails because the certificate can’t be found on the chain.

Use case: certificate of authenticity

Alice node
Gallery owner

[<AAAA>,<AAAA>]
January 1, 2017
SignedA[#CERTIFICATEHASH ABC]
Public Key Alice
Property: "owner = John Doe"

[<AAAA>,<AAAA>]
February 1, 2017
SignedB[#CERTIFICATEHASH DEF]
Public Key Bob
Property: "owner = Chuck Coe"

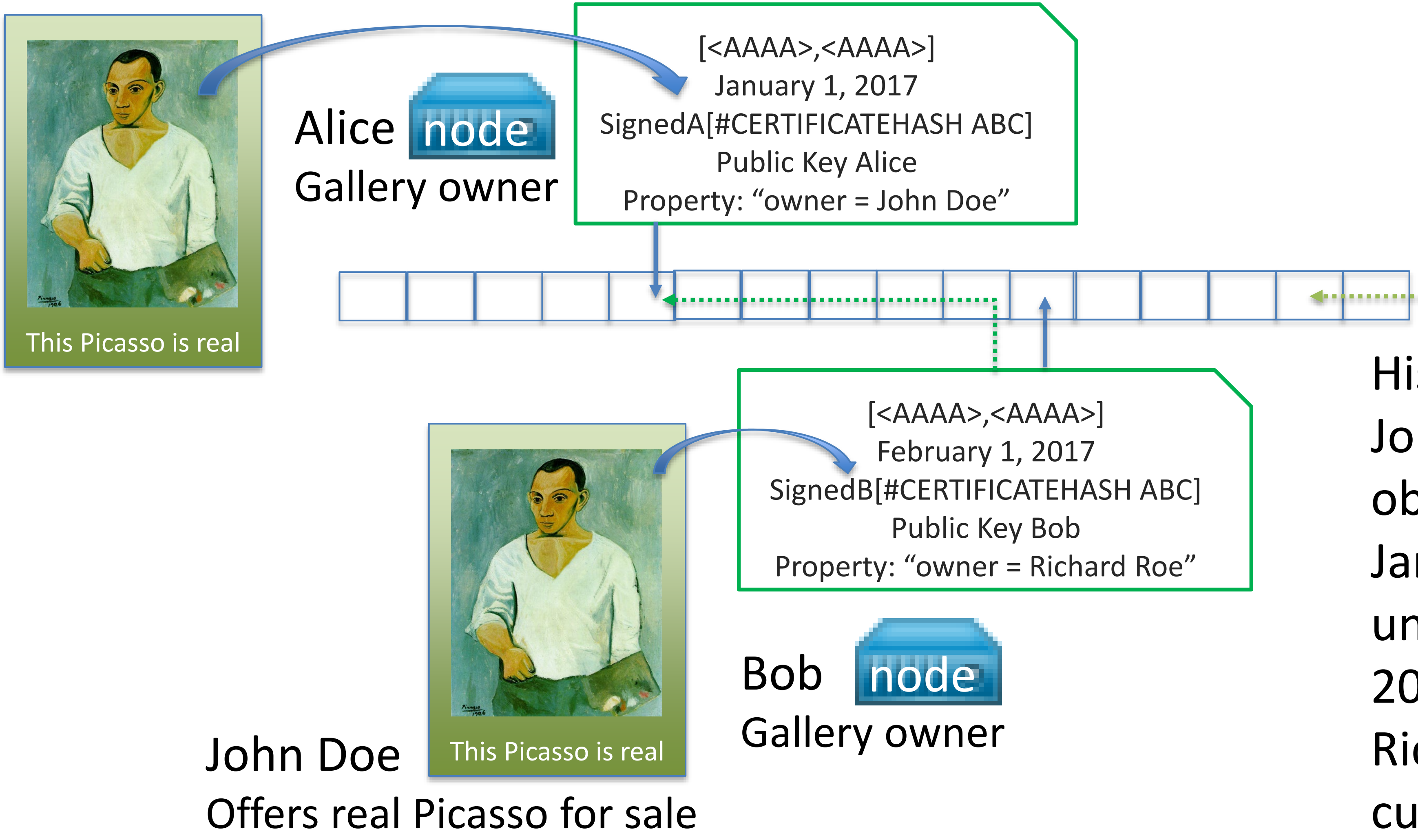
Bob node
Gallery owner

Chuck
Pretends to be John Doe
Offers fake Picasso for sale

2nd attempt to offer a forged painting with a fake certificate fails because the hashes don't correspond.

ITEXT

Use case: certificate of authenticity



History shows that John Doe owned an object of art from January 1, 2017 until February 1, 2017, and that Richard Roe is the current owner

Use case: certificate of authenticity

Alice node
Gallery owner

[<AAAA>,<AAAA>]
January 1, 2017
SignedA[#CERTIFICATEHASH ABC]
Public Key Alice
Property: “owner = John Doe”

[<AAAA>,<AAAA>]
January 1, 2017
SignedC[#CERTIFICATEHASH ABC]
Public Key Carol
Property: “owner = John Doe”

node Carol
Insurance services

Carol notices pending sale
Alerts John and Bob

[<AAAA>,<AAAA>]
February 1, 2017
SignedB[#CERTIFICATEHASH ABC]
Public Key Bob
Property: “owner = Chuck Coe”

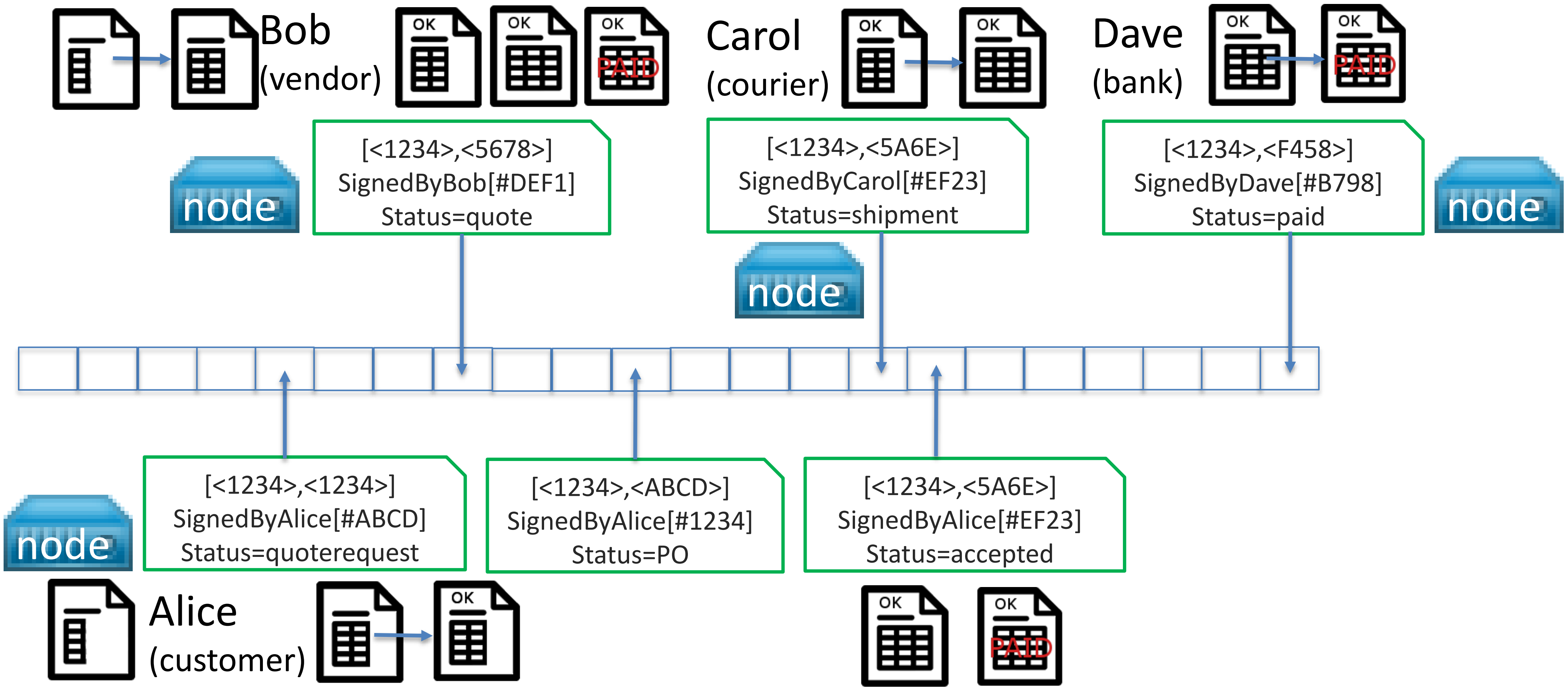
Bob node
Gallery owner

Chuck
Pretends to be John Doe
Offers fake Picasso for sale

3rd attempt to offer a forged painting with a real certificate fails because Carol monitors the certificate for John Doe.

Use case 2: Supply chain

Supply chain



Use case 3: Long-Term Validation

Renewing a signature

Bob makes it impossible for Alice to exploit flaws that are discovered in old algorithms, or to take advantage of increased processing power to forge the document and calculate a matching hash.

Alice
(debtor)

node

[<ABC123>,<ABC123>]
SignedByAlice(#CAB578FA)
Hashing Algo X

[<ABC123>,<ABC123>]
SignedByBob(#CAB578FA)
Hashing Algo X

[<ABC123>,<ABC123>]
SignedByBob(#FA358CBC01)
Hashing Algo Y

[<ABC123>,<ABC123>]
SignedByBob(#BA687EA115DF)
Hashing Algo Z

node

Bob
(creditor)

Summarized

- One public blockchain is needed
  - Strength is in the numbers
  - Separate chains: workflow can’t be tracked
- All data in the blockchain is public
  - This doesn’t mean all data needs to be public